Wanneer een bij, een bloemenveld of een bijenkorf op hetzelfde vakje staan, raken ze elkaar aan.

{\displaystyle \forall }{\displaystyle \forall }∀a ∀b ∃p ((PhysicalConcept(a) ˄ PhysicalConcept(b) ˄ Tile(p) ˄ (IsAt(a, p) ˄ IsAt(b, p)) -> Touch(a, b)))

Als een bij geen nectar heeft en op een bloemenveld staat die wel nectar heeft, pakt hij een nectar van dat bloemenveld af.

∀a ∀b ((Bee(a) ˄ FlowerField(b) ˄ ¬ HasNectar(a) ˄ (Touch(a, b) ˄ Nectar(b) ) -> (GainNectar(a) ˄ LoseNectar(b)))))

Wanneer een bij nectar heeft, wil hij naar huis gaan.

∀a ∃k ((Bee(a) ˄ Beehive(k) ˄ HasNectar(a)) -> GoHome(a, k)))

Wanneer een bij net genoeg energie heeft om naar huis te gaan, gaat hij naar huis.

∀a ∃k ((Bee(a) ˄ Beehive(k) ˄ NeedToGoHome(TicksToHome(a, k), TicksOfEnergy(a)) -> GoHome(a, b)))

Wanneer een bij de volledige grid heeft bezocht, wil hij naar huis gaan.

∀a ∃g ∃k ((Bee(a) ˄ Beehive(k) ˄ GridExplored(g)) -> GoHome(a, k)))

Wanneer een bij op zijn korf staat, verliest hij zijn grid memory. En als zijn laatst bezochte bloemenveld nog nectar bevat, krijgt hij een aanwijzing waar dit bloemenveld zich bevindt. Ook geeft de bij zijn nectar af en vult daarmee het nectar gehalte van de korf aan.

∀a ∃k ∃f ((Bee(a) ˄ Beehive(k) ˄ FlowerField(f) ˄ Touch(a, k) -> (LoseGridMemory(a) ˄ GainHoney(k) ˄ LoseNectar(a) ˄ (HasNectar(f) -> Clue(a, f))))

Wanneer een bij de korf verlaat, neemt hij een beetje honing van de korf die hij gebruikt om het energielevel van de bij volledig aan te vullen.

∀a ∃k ((Bee(a) ˄ Beehive(k) ˄ (Touch(a, k) ˄ HasHoney(k) ˄ LeavesHive(a)) -> (GainEnergy(a) ˄ LoseHoney(k))))

Bloemen gaan langzaam dood.

∀a (Flower(a) -> StartDeathCountDown(a))

Bijen gaan langzaam dood.

∀a ((Bee(a) ˄ IsZero(DeathCountDown(a)) -> Dies(a))

Saadjes gaan langzaam bloeien.

∀a (Seed(a) -> StartBirthCountDown(a))

Saadjes worden langzaam bloemen.

∀a ((Seed(a) ˄ IsZero(BirthCountDown(a)) -> BecomeFlower(a))

Wanneer bijen op bloemen staan, en ze die bestuiven, zullen de bloemen zaadjes krijgen die later uit zullen groeien tot bloemen.

∀a ∀b ∀c ((FlowerField(a) ˄ Flower(b) ˄ Bee(c) ˄ Touch(a, c) ˄ Pollinate(a, c)) -> GainSeeds(a, b))

Wanneer een bij geen energie meer heeft, gaat hij dood.

∀a ((Bee(a) ˄ NoEnergy(a)) -> Die(a))

Wanneer een bij een bloem ziet, gaat hij daarop af.

∀a ∀b ∃g ((Bee(a) ˄ FlowerField(b) ˄ ¬ Touch(a, b) ˄ Sight(a, b, g)) -> GoToFlowerfield(a))

Een vakje zit naast een ander vakje in de grid.

∀a ∀b ((Tile(a) ˄ Tile(b) ˄ (Equal(xas(a), xas(b)) ˄ (Equal(MinusOne(yas(a)), yas(b)) ∨ Equal(PlusOne(yas(a)), yas(b)))) ∨ (Equal(yas(a), yas(b)) ˄ (Equal(MinusOne(xas(a)), xas(b)) ∨ Equal(PlusOne(xas(a)), xas(b))))) -> Neighbours(a, b))

Een bij kan alleen bewegen naar een vakje naast hem op de grid.

∀a ∃b ∀c ((Bee(a) ˄ Tile(b) ˄ Tile(c) ˄ Neighbours(a, b) ˄ WantMove(a)) -> Move(a, b))